

Contents

<i>Prologue: Faultless systems – yes we can!</i>	<i>page xi</i>
<i>Acknowledgments</i>	<i>xxv</i>
1 Introduction	1
1.1 Motivation	1
1.2 Overview of the chapters	2
1.3 How to use this book	8
1.4 Formal methods	10
1.5 A little detour: blueprints	12
1.6 The requirements document	13
1.7 Definition of the term “formal method” as used in this book	16
1.8 Informal overview of discrete models	18
1.9 References	22
2 Controlling cars on a bridge	24
2.1 Introduction	24
2.2 Requirements document	25
2.3 Refinement strategy	27
2.4 Initial model: limiting the number of cars	27
2.5 First refinement: introducing the one-way bridge	50
2.6 Second refinement: introducing the traffic lights	70
2.7 Third refinement: introducing car sensors	88
3 A mechanical press controller	100
3.1 Informal description	100
3.2 Design patterns	103
3.3 Requirements of the mechanical press	114
3.4 Refinement strategy	116
3.5 Initial model: connecting the controller to the motor	117
3.6 First refinement: connecting the motor buttons to the controller	119

3.7	Second refinement: connecting the controller to the clutch	127
3.8	Another design pattern: weak synchronization of two strong reactions	127
3.9	Third refinement: constraining the clutch and the motor	135
3.10	Fourth refinement: connecting the controller to the door	137
3.11	Fifth refinement: constraining the clutch and the door	138
3.12	Another design pattern: strong synchronization of two strong reactions	139
3.13	Sixth refinement: more constraints between clutch and door	146
3.14	Seventh refinement: connecting the controller to the clutch buttons	147
4	A simple file transfer protocol	149
4.1	Requirements	149
4.2	Refinement strategy	150
4.3	Protocol initial model	151
4.4	Protocol first refinement	158
4.5	Protocol second refinement	167
4.6	Protocol third refinement	169
4.7	Development revisited	172
4.8	Reference	175
5	The Event-B modeling notation and proof obligation rules	176
5.1	The Event-B notation	176
5.2	Proof obligation rules	188
6	Bounded re-transmission protocol	204
6.1	Informal presentation of the bounded re-transmission protocol	204
6.2	Requirements document	210
6.3	Refinement strategy	211
6.4	Initial model	212
6.5	First and second refinements	213
6.6	Third refinement	215
6.7	Fourth refinement	216
6.8	Fifth refinement	221
6.9	Sixth refinement	225
6.10	Reference	226
7	Development of a concurrent program	227
7.1	Comparing distributed and concurrent programs	227
7.2	The proposed example	228
7.3	Interleaving	233
7.4	Specifying the concurrent program	237
7.5	Refinement strategy	242
7.6	First refinement	245
7.7	Second refinement	250

7.8	Third refinement	253
7.9	Fourth refinement	255
7.10	Reference	257
8	Development of electronic circuits	258
8.1	Introduction	258
8.2	A first example	267
8.3	Second example: the arbiter	280
8.4	Third example: a special road traffic light	291
8.5	The Light circuit	299
8.6	Reference	305
9	Mathematical language	306
9.1	Sequent calculus	306
9.2	The propositional language	310
9.3	The predicate language	316
9.4	Introducing equality	319
9.5	The set-theoretic language	321
9.6	Boolean and arithmetic language	331
9.7	Advanced data structures	334
10	Leader election on a ring-shaped network	353
10.1	Requirement document	353
10.2	Initial model	355
10.3	Discussion	356
10.4	First refinement	359
10.5	Proofs	361
10.6	Reference	366
11	Synchronizing a tree-shaped network	367
11.1	Introduction	367
11.2	Initial model	369
11.3	First refinement	371
11.4	Second refinement	375
11.5	Third refinement	377
11.6	Fourth refinements	384
11.7	References	386
12	Routing algorithm for a mobile agent	387
12.1	Informal description of the problem	387
12.2	Initial model	392
12.3	First refinement	396
12.4	Second refinement	399

12.5	Third refinement: data refinement	403
12.6	Fourth refinement	405
12.7	References	405
13	Leader election on a connected graph network	406
13.1	Initial model	407
13.2	First refinement	407
13.3	Second refinement	410
13.4	Third refinement: the problem of contention	412
13.5	Fourth refinement: simplification	414
13.6	Fifth refinement: introducing cardinality	415
14	Mathematical models for proof obligations	417
14.1	Introduction	417
14.2	Proof obligation rules for invariant preservation	418
14.3	Observing the evolution of discrete transition systems: traces	420
14.4	Presentation of simple refinement by means of traces	424
14.5	General refinement set-theoretic representation	431
14.6	Breaking the one-to-one relationship between abstract and concrete events	441
15	Development of sequential programs	446
15.1	A systematic approach to sequential program development	446
15.2	A very simple example	450
15.3	Merging rules	453
15.4	Example: binary search in a sorted array	454
15.5	Example: minimum of an array of natural numbers	458
15.6	Example: array partitioning	460
15.7	Example: simple sorting	463
15.8	Example: array reversing	466
15.9	Example: reversing a linked list	469
15.10	Example: simple numerical program computing the square root	473
15.11	Example: the inverse of an injective numerical function	476
16	A location access controller	481
16.1	Requirement document	481
16.2	Discussion	484
16.3	Initial model of the system	486
16.4	First refinement	488
16.5	Second refinement	492
16.6	Third refinement	497
16.7	Fourth refinement	501

17	Train system	508
17.1	Informal introduction	508
17.2	Refinement strategy	527
17.3	Initial model	528
17.4	First refinement	536
17.5	Second refinement	543
17.6	Third refinement	544
17.7	Fourth refinement	546
17.8	Conclusion	548
17.9	References	549
18	Problems	550
18.1	Exercises	551
18.2	Projects	557
18.3	Mathematical developments	572
18.4	References	582
	<i>Index</i>	584